

What is claimed is:

1 1. A method of providing server affinities for related connection request messages in
2 networking environments which perform workload balancing, comprising steps of:
3 signaling, by an executing server application, that an affinity with a selected source is to be
4 started; and
5 bypassing normal workload balancing operations, responsive to the signaling, for
6 subsequent connection request messages from the selected source while the affinity persists.

1 2. The method according to Claim 1, wherein the selected source is a selected client.

1 3. The method according to Claim 2, wherein the selected client is identified by its Internet
2 Protocol ("IP") address.

1 4. The method according to Claim 2, wherein the selected client is identified by its Internet
2 Protocol ("IP") address and port number.

1 5. The method according to Claim 1, wherein the selected source is a selected client
2 subnetwork.

1 6. The method according to Claim 1, further comprising the step of signaling, by the
2 executing server application, that the started affinity with the selected source is to be ended; and
3 wherein the step of bypassing normal workload balancing operations then ceases for subsequent

connection request messages from the selected source.

7. The method according to Claim 1, wherein the started affinity persists for a maximum duration, after which the step of bypassing normal workload balancing operations then ceases for subsequent connection request messages from the selected source.

8. The method according to Claim 7, wherein the executing server application may override the maximum duration when signaling the start of the affinity.

9. The method according to Claim 7, wherein each of the subsequent connection request messages automatically extends the maximum duration of the started affinity.

10. The method according to Claim 9, further comprising the step of extending, by the executing server application, the started affinity beyond the maximum duration.

11. The method according to Claim 1, wherein the bypassing step causes the subsequent connection request messages from the selected source to be routed to an instance of the executing server application which signaled the affinity start.

12. A method of routing related connection requests in a networking environment which performs workload balancing, comprising steps of:

storing information for enforcing one or more currently-active affinities, responsive to

receiving start affinity requests for each such currently-active affinity from one or more executing server applications;

receiving incoming connection requests from client applications; and

routing each received connection request to a proper one of the executing server applications, further comprising steps of:

selecting a particular one of the executing server applications using the stored information for enforcing affinities, when the client application sending the received connection request is identified in the stored information as having one of the currently-active affinities with the particular one; and

selecting the particular one of the executing server applications using workload balancing otherwise.

13. The method according to Claim 12, wherein the client application is identified as having one of the currently-active affinities with the particular one if a destination address and destination port, as well as a source address and optionally a source port number, of the connection request being routed match the stored information.

14. The method according to Claim 12, further comprising the step of removing stored information for enforcing selected ones of the currently-active affinities, responsive to receiving an end affinity request from selected ones of the executing server applications which stored the information.

1 15. The method according to Claim 12, further comprising the step of removing stored
2 information for enforcing selected ones of the currently-active affinities, responsive to expiration
3 of a duration value for the selected ones.

1 16. A system for providing server affinities for related connection request messages in
2 networking environments which perform workload balancing, comprising:

3 means for signaling, by an executing server application, that an affinity with a selected
4 source is to be started; and

5 means for bypassing normal workload balancing operations, responsive to the signaling,
6 for subsequent connection request messages from the selected source while the affinity persists.

1 17. The system according to Claim 16, further comprising means for signaling, by the
2 executing server application, that the started affinity with the selected source is to be ended; and
3 wherein the means for bypassing normal workload balancing operations then ceases for
4 subsequent connection request messages from the selected source.

1 18. The system according to Claim 16, wherein the started affinity persists for a maximum
2 duration, after which the means for bypassing normal workload balancing operations then ceases
3 for subsequent connection request messages from the selected source.

1 19. The system according to Claim 18, wherein the executing server application may override
2 the maximum duration when signaling the start of the affinity.

1 20. The system according to Claim 18, wherein each of the subsequent connection request
2 messages automatically extends the maximum duration of the started affinity.

1 21. The system according to Claim 20, further comprising means for extending, by the
2 executing server application, the started affinity beyond the maximum duration.

1 22. The system according to Claim 16, wherein the means for bypassing causes the subsequent
2 connection request messages from the selected source to be routed to an instance of the executing
3 server application which signaled the affinity start.

1 23. A system for routing related connection requests in a networking environment which
2 performs workload balancing, comprising:

3 means for storing information for enforcing one or more currently-active affinities,
4 responsive to receiving start affinity requests for each such currently-active affinity from one or
5 more executing server applications;

6 means for receiving incoming connection requests from client applications; and

7 means for routing each received connection request to a proper one of the executing
8 server applications, further comprising:

9 means for selecting a particular one of the executing server applications using the
10 stored information for enforcing affinities, when the client application sending the received
11 connection request is identified in the stored information as having one of the currently-active

12 affinities with the particular one; and

13 means for selecting the particular one of the executing server applications using
14 workload balancing otherwise.

1 24. The system according to Claim 23, wherein the client application is identified as having
2 one of the currently-active affinities with the particular one if a destination address and destination
3 port, as well as a source address and optionally a source port number, of the connection request
4 being routed match the stored information.

1 25. The system according to Claim 23, further comprising means for removing stored
2 information for enforcing selected ones of the currently-active affinities, responsive to receiving
3 an end affinity request from selected ones of the executing server applications which stored the
4 information.

1 26. The system according to Claim 23, further comprising means for removing stored
2 information for enforcing selected ones of the currently-active affinities, responsive to expiration
3 of a duration value for the selected ones.

1 27. A computer program product for providing server affinities for related connection request
2 messages in networking environments which perform workload balancing, the computer program
3 product embodied on one or more computer readable media and comprising:

4 computer readable program code means for signaling, by an executing server application,

that an affinity with a selected source is to be started; and

computer readable program code means for bypassing normal workload balancing operations, responsive to the signaling, for subsequent connection request messages from the selected source while the affinity persists.

28. The computer program product according to Claim 27, further comprising computer readable program code means for signaling, by the executing server application, that the started affinity with the selected source is to be ended; and wherein the computer readable program code means for bypassing normal workload balancing operations then ceases for subsequent connection request messages from the selected source.

29. The computer program product according to Claim 27, wherein the started affinity persists for a maximum duration, after which the computer readable program code means for bypassing normal workload balancing operations then ceases for subsequent connection request messages from the selected source.

30. The computer program product according to Claim 29, wherein the executing server application may override the maximum duration when signaling the start of the affinity.

31. The computer program product according to Claim 29, wherein each of the subsequent connection request messages automatically extends the maximum duration of the started affinity.

1 32. The computer program product according to Claim 30, further comprising computer
2 readable program code means for extending, by the executing server application, the started
3 affinity beyond the maximum duration.

1 33. The computer program product according to Claim 27, wherein the computer readable
2 program code means for bypassing causes the subsequent connection request messages from the
3 selected source to be routed to an instance of the executing server application which signaled the
4 affinity start.

1 34. A computer program product for routing related connection requests in a networking
2 environment which performs workload balancing, the computer program product embodied on
3 one or more computer readable media and comprising:

4 computer readable program code means for storing information for enforcing one or more
5 currently-active affinities, responsive to receiving start affinity requests for each such currently-
6 active affinity from one or more executing server applications;

7 computer readable program code means for receiving incoming connection requests from
8 client applications; and

9 computer readable program code means for routing each received connection request to a
10 proper one of the executing server applications, further comprising:

11 computer readable program code means for selecting a particular one of the
12 executing server applications using the stored information for enforcing affinities, when the client
13 application sending the received connection request is identified in the stored information as

14 having one of the currently-active affinities with the particular one; and

15 computer readable program code means for selecting the particular one of the
16 executing server applications using workload balancing otherwise.

1 35. The computer program product according to Claim 34, wherein the client application is
2 identified as having one of the currently-active affinities with the particular one if a destination
3 address and destination port, as well as a source address and optionally a source port number, of
4 the connection request being routed match the stored information.

1 36. The computer program product according to Claim 34, further comprising computer
2 readable program code means for removing stored information for enforcing selected ones of the
3 currently-active affinities, responsive to receiving an end affinity request from selected ones of the
4 executing server applications which stored the information.

1 37. The computer program product according to Claim 34, further comprising computer
2 readable program code means for removing stored information for enforcing selected ones of the
3 currently-active affinities, responsive to expiration of a duration value for the selected ones.